Hierarchical expectation propagation for Bayesian aggregation of average data[*]

Sebastian Weber[†]   Andrew Gelman[‡]   Bob Carpenter [‡]   Daniel Lee[‡]
Michael Betancourt[§]   Aki Vehtari[¶]   Amy Racine[†]
26 Oct 2015

Abstract

We often wish to use external data to improve the precision of an inference, but concerns arise when the different datasets have been collected under different conditions so that we do not want to simply pool the information. This is the well-known problem of *meta-analysis*, for which Bayesian methods have long been used to achieve partial pooling. Here we consider the challenge when the external data are averages rather than raw data. We provide a Bayesian solution by using simulation to approximate the likelihood of the external summary, and by allowing the parameters in the model to vary under the different conditions. Inferences are constructed using importance sampling from an approximate distribution determined by an expectation propagation-like algorithm. We demonstrate with the problem that motivated this research, a hierarchical nonlinear model in pharmacometrics, implementing the computation in Stan.

Keywords: Bayesian computation, Expectation propagation, Hierarchical modeling, Importance sampling, Meta-analysis, Pharmacometrics, Stan

1.   The method

This paper describes a statistical computational approach for integrating averaged data from an external source into a hierarchical Bayesian analysis, taking account of possible differences in the model in the two datasets. In this section we describe the algorithm; section 2 demonstrates on some examples with simulated data; and section 3 applies the method to the problem that motivated this work, a nonlinear differential equation model in pharmacometrics.

1.1.   General formulation

We shall work in a hierarchical Bayesian framework. Suppose we have data $y = (y_{jt}; j = 1, \ldots, J; t = 1, \ldots, T)$ on J individuals at T time points, where each $y_j = (y_{j1}, \ldots, y_{jT})$ is a vector of data with model $p(y_j | \alpha_j, \phi)$. Here, each α_j is a vector of parameters for individual j, and ϕ is a vector of shared parameters and hyperparameters, so that the joint prior is $p(\alpha, \phi) = p(\phi) \prod_{j=1}^{J} p(\alpha_j | \phi)$, and the primary goal of the analysis is inference for the parameter vector ϕ.

We assume that we can use an existing computer program such as Stan (Stan Development Team, 2015) to draw posterior simulations from $p(\alpha, \phi | y) \propto p(\phi) \prod_{j=1}^{J} p(\alpha_j | \phi) \prod_{j=1}^{J} p(y_j | \alpha_j, \phi)$.

We then want to update our inference using an *external dataset*, $y' = (y'_{jt}; j = 1, \ldots, J'; t = 1, \ldots, T')$, on J' individuals at T' time points, assumed to be generated under the model, $p(y'_j | \alpha'_j, \phi')$. There are two complications:

- The external data, y', are modeled using a process with parameters ϕ' that are similar to but not identical to those of the original data. We shall express our model in terms of the difference between the two parameter vectors, $\delta = \phi' - \phi$. We assume the prior distribution factorizes as $p(\phi, \delta) = p(\phi)p(\delta)$.

  We assume that all the differences between the two studies, and the populations which they represent, are captured in δ. One could think of ϕ and ϕ' as two instances from a population of studies; if we were to combine data from several external studies it would make sense to include between-study variation using an additional set of hyperparameters in the hierarchical model.

- We do not measure y' directly; instead we observe the time series of averages, $\bar{y}' = (\bar{y}'_1, \ldots, \bar{y}'_T)$. And, because of nonlinearity in the data model, we cannot simply write the model for the external average data, $p(\bar{y}'|\alpha', \phi')$, in closed form.

This is a problem of meta-analysis, for which there is a longstanding concern when the different pieces of information to be combined come from different sources or are reported in different ways (see, for example, Higgins and Whitehead, 1996; Dominici et al., 1999).

The two data issues listed above lead to corresponding statistical difficulties:

- If the parameters ϕ' of the external data were completely unrelated to the parameters of interest, ϕ—that is, if we had a noninformative prior distribution on their difference, δ—then there would be no gain to including the external data into the model, assuming the goal is to learn about ϕ.

  Conversely, if the two parameter vectors were identical, so that $\delta \equiv 0$, then we could just pool the two datasets. The difficulty arises because the information is partially shared, to an extent governed by the prior distribution on δ.

- Given that we see only averages of the external data, the conceptually simplest way to proceed would be to consider the individual measurements y'_{jt} as missing data, and to perform Bayesian inference jointly on all unknowns, obtaining draws from the posterior distribution, $p(\phi, \delta, \alpha'|y, \bar{y}')$. The difficulty here is computational: every missing data point adds to the dimensionality of the joint posterior distribution, and the missing data can be poorly identified from the model and the average data; weak data in a nonlinear model can lead to a poorly-regularized posterior distribution that is hard to sample from.

As noted, we resolve the first difficulty using an informative prior distribution on δ. We resolve the second difficulty via a normal approximation, taking advantage of the fact that our observed data summaries are averages.

1.2. Basic importance sampling algorithm

Our basic idea is to approximate the probability model for the aggregate data, $p(\bar{y}'|\phi')$, by a multivariate normal whose parameters depend on $\bar{y}'$—an approximation which is justified from the central limit theorem if the summary is an average over many data points—and then to use importance weighting to update $p(\phi|y)$ to account for the likelihood of the external data.

The algorithm proceeds as follows:

1. Perform Bayesian inference using the original data y and obtain S posterior simulation draws, $(\alpha_s, \phi_s), s = 1, \ldots, S$, from $p(\alpha, \phi|y)$. We will discard the α's and just use the simulated ϕ's.

2. For each ϕ_s:

   (a) Sample one draw of δ from its prior, $p(\delta)$, and call it δ_s. This, along with ϕ_s, can be thought of as a random draw from the joint posterior distribution, $p(\phi, \delta | y)$.

   (b) Compute $\phi'^s = \phi_s + \delta_s$

   (c) Simulate parameters $\tilde{\alpha}_j$ and then data $\tilde{y}_{jt}, j = 1, \ldots, \tilde{J}, t = 1, \ldots, T'$, for some number of hypothetical new individuals (for example, $\tilde{J} = 100$), drawn from the distribution $p(y|\phi'^s)$, corresponding to the conditions under which the external data were collected (hence the use of the same number of time points T'). The $\tilde{J}$ individuals do *not* correspond to the J' individuals in the external dataset; rather, we simulate them only as a device for approximating the likelihood of average data, $\bar{y}$, under these conditions.

   (d) Compute the mean vector and the $T' \times T'$ covariance matrix of the simulated data $\tilde{y}$. Call these $\tilde{M}_s$ and $\tilde{\Sigma}_s$.

   (e) Divide the covariance matrix $\tilde{\Sigma}_s$ by J' to get the simulation-estimated covariance matrix for $\bar{y}'$, which is an average over J' individuals whose data are modeled as independent conditional on the parameter vector ϕ'.

   (f) Approximate the probability density of the observed mean vector of the J' external data points using the multivariate normal distribution with mean $\tilde{M}_s$ and covariance matrix $\frac{1}{J'}\tilde{\Sigma}_s$, and use this density as an *importance ratio*, $r_s = \mathrm{N}(\bar{y}'|\tilde{M}_s, \frac{1}{J'}\tilde{\Sigma}_s)$, representing the information from the external data.

3. Compute the *Pareto-smoothed importance weights* w_s, which are constructed from the raw ratios r_s (for example using the function `psislw()` in the R package `loo`) to stabilize inferences when importance weights are highly variable (Vehtari et al., 2015). In the three examples of the present paper, the algorithm performed better using Pareto smoothing than when using raw ratios or simple truncation.

4. Compute posterior expectations of the parameter vectors ϕ and δ using the S importance-weighted simulations. If approximate draws from the posterior distribution are desired, subsample using importance resampling.

1.3. Pseudo-priors for ϕ and δ

The algorithm of section 1.2 has a weakness and that is the sampling of δ from its prior distribution. For one thing, this step is impossible if δ has an improper prior distribution. More generally, the algorithm can be improved if we allow tuning of the distribution from which δ is drawn. Once we are doing this, we can also use an approximate distribution for ϕ.

Our more general algorithm uses a *pseudo-prior*, $g(\phi, \delta)$, which we insert into our fitting of the model to data y in step 1 of the above algorithm. The pseudo-prior is intended to capture, as well as possible, all the information in the problem *except* that from the likelihood of the original data, $p(y|\alpha, \phi)$, so that the resulting importance sampling adjustment will be minimal. The inference for the pseudo-prior parameters is part of the algorithm.

For simplicity we assume the pseudo-prior is Gaussian and that it factors into independent pieces for ϕ and δ, so that $g(\phi, \delta) = g(\phi)g(\delta)$. We can consider relaxing these conditions, but for now we emphasize that they do not represent parametric assumptions on the model itself; rather, they are restrictions that limit the accuracy of the approximation and thus put more of a burden on the importance-weighting correction that occurs at the end of the algorithm.

The algorithm above is altered in three places:

- In step 1, perform Bayesian inference on $g(\phi)p(\alpha|\phi)p(y|\alpha,\phi)$; that is, replace the prior $p(\phi)$ in the model by the pseudo-prior $g(\phi)$.

- In step 2a, draw δ from its pseudo-prior $g(\delta)$ rather than from the prior $p(\delta)$ in the model.

- Replace the importance ratio in step 2f by:

$$r_s = \frac{p(\phi_s)p(\delta_s|\phi_s)}{g(\phi_s)g(\delta_s)} \mathrm{N}(\bar{y}'|\tilde{M}_s, \frac{1}{J'}\tilde{\Sigma}_s). \tag{1}$$

1.4. Using expectation propagation to improve the importance sampling approximation

The improvement described in the previous section should perform best if the pseudo-prior captures the information in the model so that the importance weights in (1) are close to constant. But we do not know how to construct such an approximation until we have done some sampling. This dilemma motivates an iterative procedure, in which we start by performing the algorithm described in section 1.3 above, using some vague but proper $g(\delta)$ as a starting point and then use the resulting simulated values δ_s and smoothed importance weights w_s to construct good pseudo-priors, that is, infer parameters of the pseudo-priors.

Obtaining a pseudo-prior for δ is easy in this setup, because we chose $g(\delta)$ to be Gaussian and all the information about δ comes from the second part of the model. We simply take the output from step 4 of the above algorithm and compute the weighted mean and covariance matrix of the posterior simulations of δ:

$$\bar{\delta} = \frac{\sum_{s=1}^{s} w_s \delta_s}{\sum_{s=1}^{s} w_s} \quad \text{and} \quad V_\delta = \frac{\sum_{s=1}^{s} w_s(\delta_s - \bar{\delta})(\delta_s - \bar{\delta})^t}{\sum_{s=1}^{s} w_s}, \tag{2}$$

and set $g(\delta)$ to $\mathrm{N}(\bar{\delta}, V_\delta)$ for the next iteration of the algorithm. This can be seen as a version of expectation propagation (EP; Minka, 2001; Opper and Winther, 2000, 2005). EP is typically used to construct a distributional approximation, but it can also be combined with MCMC and importance sampling (Gelman et al., 2014).

The construction of the pseudo-prior for ϕ is more involved as we need to take the posterior distribution for ϕ given all the data, and "subtract" the information from the first part of the model. We do this using an algorithm inspired by expectation propagation.

The key step of an EP-like algorithm is to approximate with a cavity distribution a contribution to the likelihood, not in isolation but in the context of the rest of the posterior distribution. For the importance sampling algorithm of the present article, the steps go as follows.

We define three distributions on ϕ:

- $p_0 \propto g(\phi)$, the pseudo-prior used in the inference for ϕ, which for convenience we label here as $\mathrm{N}(\phi|\mu_0, \Sigma_0)$.

- $p_1 \propto g(\phi)p(y|\phi)$, the *pseudo-posterior* after observing the data y, which we approximate as a Gaussian and label $\mathrm{N}(\phi|\mu_1, \Sigma_1)$.

- $p_2 \propto \int p(\phi)p(y|\phi)p(\delta)p(\bar{y}'|\phi+\delta)d\delta$, the pseudo-posterior, which we compute not by performing the integral but by getting approximate draws from the joint posterior of (ϕ, δ), just looking at the draws of ϕ and approximating as a Gaussian, which we label $\mathrm{N}(\phi|\mu_2, \Sigma_2)$.

We obtain the Gaussian approximations as follows:

- We are already using a Gaussian for p_0, the pseudo-prior for ϕ.

- p_1 is simply the distribution computed using the sample mean and covariance of the posterior simulations of ϕ obtained in step 1 of the above algorithm as altered in section 1.3.

- p_2 is what is obtained after importance-weighting the draws from p_1, so we get the normal approximation from the weighted mean and weighted covariance of the posterior simulations using the same formulas as (2) but applied to ϕ rather than δ.

We can use the above terms to compute $p_0 p_2 / p_1$, which represents all the information about ϕ in the model *other than* from the direct data likelihood $p(y|\phi)$. We compute the mean and variance of the normal approximation $N(\phi|\mu_{2-1}, \Sigma_{2-1})$ for $p_0 p_2 / p_1$ as

$$
\begin{aligned}
\Sigma_{2-1}^{-1} &= \Sigma_0^{-1} + \Sigma_2^{-1} - \Sigma_1^{-1} \\
\widehat{\Sigma}_{2-1}^{-1} \mu_{2-1} &= \Sigma_0^{-1} \mu_0 + \Sigma_2^{-1} \mu_2 - \Sigma_1^{-1} \mu_1,
\end{aligned}
\tag{3}
$$

and we update the pseudo-prior $g(\phi)$ to $N(\mu_{2-1}, \Sigma_{2-1})$. Ideally it could make more sense to apply EP on the joint distribution of (ϕ, δ); here we are computing independent pseudo-priors, $g(\delta)$ and $g(\phi)$, for simplicity, taking advantage of the structure of this problem in which only one of the two data sources supplies information about ϕ.

The covariance matrix in (3) may require a correction to ensure that the resulting covariance matrix is positive definite. This stems from the fact that the normal distribution is only an approximation to the posterior distribution, and there is also simulation variability, so the matrix Σ' as computed above might not be positive definite. In such cases, various solutions have been considered in the EP literature. Here we shall simply perform a relaxation where we keep halving the jump size until the covariance matrix is positive definite; that is,

$$
\begin{aligned}
\widehat{\Sigma}_{2-1}^{-1} &= \Sigma_0^{-1} + \Sigma_2^{-1} - \frac{1}{2^n} \Sigma_2^{-1} \\
\widehat{\Sigma}_{2-1}^{-1} \hat{\mu}_{2-1} &= \Sigma_0^{-1} \mu_0 + \Sigma_2^{-1} \mu_2 - \frac{1}{2^n} \Sigma_2^{-1} \mu_2,
\end{aligned}
$$

where n is the smallest nonnegative integer for which $\widehat{\Sigma}_{2-1}^{-1}$ is positive definite.

1.5. Putting the pieces together

We are now ready to insert the EP approximations into an algorithm that iterates over the pseudo-prior to obtain a better approximation and thus more stable importance weights.

The algorithm proceeds as follows:

0. Start with a vague choice of pseudo-priors on ϕ and δ, for example independent $N(0, 1)$ densities on all parameters (assuming they have been transformed to an unbounded scale, for example by taking logarithms of all-positive parameters and logits of parameters with interval constraints).

1. Perform inference for the posterior density $g(\alpha, \phi|y) \propto g(\phi) \prod_{j=1}^{J} p(\alpha_j|\phi) \prod_{j=1}^{J} p(y_j|\alpha_j, \phi)$. Discard the simulations of α, and label the simulations of ϕ as $\phi_s, s = 1, \ldots, S$.

2. For each ϕ_s:

   (a) Sample one draw of δ from its pseudo-prior, $g(\delta)$, and call it δ_s. This, along with ϕ_s, can be thought of as a random draw from the joint pseudo-posterior distribution, $g(\phi, \delta|y)$.

   (b) Compute $\phi'^s = \phi_s + \delta_s$

(c) Simulate parameters $\tilde{\alpha}_j$ and then data $\tilde{y}_{jt}; j = 1, \ldots, \tilde{J}; t = 1, \ldots, T'$ for some number of hypothetical new individuals (for example, $\tilde{J} = 100$), drawn from the distribution $p(y|\phi'^s)$, corresponding to the conditions under which the external data were collected.

(d) Compute the mean vector and the $T' \times T'$ covariance matrix of the simulated data $\tilde{y}$. Call these $\tilde{M}_s$ and $\tilde{\Sigma}_s$.

(e) Divide the covariance matrix $\tilde{\Sigma}_s$ by J' to get the simulation-estimated covariance matrix for $\bar{y}'$, which is an average over J' individuals whose data are modeled as independent conditional on the parameter vector ϕ'.

(f) Approximate the probability density of the observed mean vector of the J' external data points using the multivariate normal distribution with mean $\tilde{M}_s$ and covariance matrix $\frac{1}{J'}\tilde{\Sigma}_s$.

(g) Now that we have this approximation to the likelihood for $\bar{y}'$, compute the importance ratios,

$$
\begin{aligned}
r_s^{(1)} &= \frac{p(\phi_s)}{g(\phi_s)} \\
r_s^{(2)} &= \frac{p(\delta_s)}{g(\delta_s)} \quad (4) \\
r_s^{(3)} &= \mathrm{N}(\bar{y}'|\tilde{M}_s, \frac{1}{J'}\tilde{\Sigma}_s). \quad (5)
\end{aligned}
$$

The first of these ratios corrects for the model used in the inference for ϕ, the second ratio corrects for the model used in the simulation of δ given ϕ, and the final factor represents the information in the external data $\bar{y}'$. We also compute their product,

$$
r_s = r_s^{(1)} r_s^{(2)} r_s^{(3)},
$$

for each simulation draw s.

(h) If the variance of the importance ratios is finite (see Section 1.5), we can roughly estimate the efficiency of the set of importance ratios using the formula,

$$
\text{efficiency} = \frac{S}{\sum_{s=1}^{S} (r_s/\bar{r})^2}, \quad (6)
$$

which equals 1 if all the weights are equal and approaches $1/S$ in the limit that one weight is huge and the others are all negligible.

3. Compute the Pareto smoothed importance weights, which are constructed from the raw ratios r_s (for example using the function `psislw()` in the R package `loo`) to stabilize inferences when importance weights are highly variable (Vehtari et al., 2015). The reliability of the importance sampling can evaluated by examining the generalized Pareto distribution shape parameter estimate $\hat{k}$ (Vehtari et al., 2015).

   - If the shape parameter k is less than $\frac{1}{2}$, then the distribution of importance ratios has finite variance, the central limit theorem holds and the importance sampling estimate convergence is fast.

   - If the shape parameter k is between $\frac{1}{2}$ and 1, then the variance is infinite, but the mean exists. The generalized central limit theorem for stable distributions holds, and the distribution of the estimate converges to a stable distribution. The convergence of importance sampling estimate is slower, the closer the shape parameter k is to 1.

- If the shape parameter k is 1 or greater, the variance and the mean of the raw importance ratios do not exist. The Pareto smoothed importance sampling estimate has a finite variance but it is likely to have some bias which is likely to cause underestimation of the uncertainties in the true posterior.

- There is the possibility that in the early stages of the iteration the approximation will be far from the target and the shape parameter k will be much larger than 1. Then it is more robust to use sampling without replacement in the early iterations and switch to Pareto smoothed importance sampling in the later iterations (see Section 1.6).

- We do not observe the shape parameter k directly, but it is estimated using an efficient procedure described by Zhang and Stephens (2009). Vehtari et al. (2015) illustrate the variation of the estimate in the case of importance ratios. We could also compute the posterior distribution for k, but at least with large sample sizes as used here, $\hat{k}$ is accurate enough and the additional information in the full posterior does not seem to help to predict the reliability of the Pareto smoothed importance sampling.

4. Do the EP computation for δ by updating the pseudo-prior $g(\delta)$ to $N(\delta|\bar{\delta}, V_\delta)$ as computed from (2).

5. Repeat steps 2–4 to obtain a better pseudo-prior for δ. These steps do not require re-fitting the original model for ϕ; they just require new draws of δ and recomputation of the importance weights.

6. Do the EP computation for ϕ:

   (a) Compute the mean and covariance of the simulations of ϕ and label these as μ_1, Σ_1

   (b) Using the latest set of weights w_s from step 3, compute the weighted mean and covariance of the simulations of ϕ; call these μ_2 and Σ_2.

   (c) Update the pseudo-prior $g(\phi)$ to $N(\phi|\mu_0, \Sigma_0)$, as computed from $N(\phi|\mu_{2-1}, \Sigma_{2-1})$ (3).

7. Return to step 1 above and repeat until approximate convergence (which we expect should just take a few of steps, not because the algorithm is perfect but because there is only so much room for improvement in the normal approximation). If the Pareto shape parameter k is 1 or greater in the end, the results should not be trusted and an improved proposal distribution in importance sampling should be used as discussed later.

1.6. Tuning and convergence

We can monitor convergence of the algorithm by running it multiple times from different starting points and then running until, for all quantities of interest, the ratio of between to within variances is close to 1 (Gelman and Rubin, 1992).

As the algorithm proceeds iterating, the importance weights are expected to stabilize. The stabilization of the importance weights can directly be monitored by observing the estimated Pareto shape $\hat{k}$ which is calculated in step 3 to smooth the importance weights. A decreasing Pareto shape $\hat{k}$ indicates a stabilization of the importance weights.

However, as we start possibly very far from target region of interest (the posterior), the above reweighing scheme may become unstable (Pareto shape $\hat{k} \gg 1$). Since we use a finite number of samples, it may occur that only very few draws are sampled from the target region while most others are very far away (in units of the residual error). In the most extreme case only a single draw has non-negligible weight. This situation may occur in the beginning of the algorithm and cause

the reweighing scheme (2) to collapse as the variance artificially vanishes in that case. To avoid this issue, we use for the first steps of the algorithm a resampling without replacement scheme to update the pseudo prior $g(\delta)$. Moreover, we ensure after step 6 that the variances for the pseudo-prior $g(\phi)$ are not decreased too fast. That is, we calculate approximately how many observations the pseudo-prior $g(\phi)$ is equivalent with respect to the prior, which we consider to be equivalent to 1 observation. Considering the number of observations to be proportional to inverse variances we obtain $n_{g(\phi)} \approx \text{var}(p(\phi))/\text{var}(g(\phi))$. Next, to the covariance matrix of $g(\phi)$ we add a diagonal covariance matrix Σ_0 so that $n_{g(\phi)}$ is no larger than a pre-specified n which we increase while the outer loop of the algorithm proceeds. This effectively limits the variance from below in the beginning of the algorithm. Since we increase n to large values with increasing iteration, this lower limit vanishes.

Even if Pareto shape $\hat{k} > 1$ the Pareto smoothed importance sampling estimate is finite and the overall algorithm can converge. If in the approximate convergence Pareto shape $\hat{k} > 1$, the results should not be trusted. The importance sampling can be improved by adapting the proposal distribution until $\hat{k} < 1$, for example, by using split-normal (Geweke, 1989) or other adaptive methods, but we leave these experiments for a future work. Using non-Gaussian proposals could be used to improve the accuracy of the importance sampling, but we are still using Gaussian messages in the expectation propagation which may cause errors in the inference if the posterior is highly non-Gaussian. We recommend using the posterior predictive checking and cross-validation to assess the overall inference results (as discussed in Gelman et al., 2013, Ch. 6–7).

The algorithm requires the following scheduling parameters. Going through the steps of the algorithm in Section 1.5: we need the number of iterations and number of chains for the fit to the data y in step 1; the number of hypothetical replications $\tilde{J}$ in step 2c; the number of steps in the looping of the pseudo-prior distribution for δ in step 5; and, finally, the number of loops for the pseudo-prior distribution for ϕ in step 7.

This is a stochastic optimization algorithm, so a further complication is that it makes sense to increase the number of internal iterations as the outer iterations proceed. For the first steps of the EP algorithm it does not make sense to waste a lot of cycles getting a very precise simulation of a distribution that is far from the target. Later, when the algorithm is close to convergence, is the time to increase the number of simulation draws.

For the particular problems we have studied, it has been feasible to tune all these parameters by hand. To implement the algorithm more automatically, it would be appropriate to get a sense of the computational bottlenecks.

2.   Simple examples

We demonstrate the algorithm in three examples with simulated data. The first example, hierarchical linear regression, is simple enough that we can easily compare our approximate inferences with simulations from the full posterior distribution under the model. We then repeat the calculation but changing the data model from linear to logistic. Last, we apply our procedure to a non-linear mixed-effect model frequently used in pharmacometrics.

2.1.   Hierarchical linear regression

We begin with a linear regression in which, for simplicity, the intercept varies by individual and the slope varies by group. That is, for the main dataset y, the model is $y_{jt} \sim \text{N}(\alpha_{j1} + \alpha_{j2}x_t + \beta x_t^2, \sigma_y^2)$, with prior distribution $\alpha_j \sim \text{N}(\mu_\alpha, \Sigma_\alpha)$. Using the notation from Section 1.1, the vector of shared

parameters α is $\phi = (\beta, \mu_\alpha, \sigma_\alpha, \sigma_y)$; the model is simple enough and we assume the number of individuals J is large enough that we can simply assign a uniform prior to ϕ.

For the external dataset y', the model is $y_{jt} \sim N(\alpha'_{j1} + \alpha'_{j2}x_t + \beta x_t^2, \sigma_y^2)$, with the prior distribution $\alpha'_j \sim N(\mu'_\alpha, \Sigma_\alpha)$. In this simple example, we assign a noninformative uniform prior distribution to $\delta = \mu'_\alpha - \mu_\alpha$.

Assumed parameter values. We create simulations assuming the following conditions, which we set to roughly correspond to the features of the pharmacometrics example in Section 3:

- $J = 50$ individuals in the original dataset, each measured $T = 13$ times (corresponding to measurements once per month for a year), $x_t = 0, \frac{1}{12}, \ldots, 1$.

- $J' = 200$ individuals in the external dataset, also measured at these 13 time points.

- $(\mu_\alpha, \sigma_\alpha)_1 = (0.5, 0.1)$, corresponding to intercepts that are mostly between 0.4 and 0.6. The data from our actual experiment roughly fell on a 100-point scale, which we are rescaling to 0–1 following the general principle in Bayesian analysis to put data and parameters on a unit scale (Gelman, 2004).

- $(\mu_\alpha, \sigma_\alpha)_2 = (-0.2, 0.1)$, corresponding to an expected loss of between 10 and 30 points on the 100-point scale for most people during the year of the study.

- $\beta = -0.1$, corresponding to an accelerating decline representing an additional drop of 10 points over the one-year period.

- $\sigma_y = 0.05$, indicating a measurement and modeling error on any observation of about 5 points on the original scale of the data.

Finally, we set δ to $(0.1, 0.1)$, which represents a large difference between groups in the context of this problem, and allows us to test how well the method works when the shift in parameters needs to be discovered from data.

In our inferences, we assign independent unit normal priors for all the parameters μ_1, μ_2, β, $\log \sigma_1$, $\log \sigma_2$, δ_1, δ_2. Given the scale of the problem, these is a weak prior which just serves to keep the inferences within generally reasonable bounds.

Conditions of the simulations. We run our algorithm 3 times to allow us to see it develop from different starting values for the pseudo-prior distributions. We initialize the pseudo-priors to be multivariate normal with identity covariance matrices (which is consistent with the weakly informative priors in the model) with the means of these pseudo-priors drawn from independent normal distributions with mean 0 and standard deviation 0.5. We want the initial pseudo-priors to be different to facilitate monitoring the convergence of the algorithm.

We fit the model in step 2 in Stan, running 4 chains with a number of iterations that increases as the algorithm proceeds, starting at 100 and then increasing by a factor of $\sqrt{2}$ at each step of the outer loop of the EP updating for ϕ. Inside each step of this loop, we run the EP updating for δ for 10 steps, and in each of these steps, we set the number of hypothetical replications of $\tilde{y}$ to 1000 to approximate the normal distribution for the distribution of the external data averages.

During the first 25 steps of the algorithm we use resampling in place of reweighting to avoid collapsing of weights onto very few draws as explained before. The maximal sample size n for $n_{g(\phi)}$ is set initially to 2 and is increased by a factor of $\sqrt{2}$ with each increment of the outer loop. Finally,

we run the outer loop for 10 steps, which is sufficient to obtain approximate convergence in our example.

The above conditions are motivated from experience with stochastic algorithms but cannot be ideal; in particular, the number of updates of δ and the number of replications of $\tilde{y}$ should also increase as the main looping proceeds.

Computation and results. We simulate data y and y'. For simplicity we do our computations just once in order to focus on our method only. If we wanted to evaluate the statistical properties of the examples, we could nest all this in a larger simulation study.

We run the algorithm as described below and reach approximate convergence in that $\widehat{R}$ is near 1 for all the parameters in the model. We then compare our inferences to three alternatives:

- The posterior mean estimates for the shared parameters ϕ using just the model fit to the local data y;

- The estimates for all the parameters ϕ, δ using the complete data y, y', which would not in general be available—from the statement of the problem we see only the averages for the new data—but we can do here as we have simulated data.

- The estimates for all the parameters ϕ, δ using the actual available data $y, \bar{y}'$. In general it would not be possible to compute this inference directly, as we need the probability density for the averaged data, but in this linear model this distribution has a simple closed-form expression which we can calculate.

Figure 1 shows the results. What we would like for each parameter is for the approximations (shown as black lines on the plots) to match the *blue line*, which correspond to the correct Bayes estimate, $p(\phi, \delta | y, \bar{y}')$. And the black lines do converge to the blue line, for each parameter. As the plots show, the algorithm did not converge immediately; it took a few steps for the pseudo-priors to get close enough.

In a real example the red lines would be available but not the blue or green lines, so the way to check convergence is to see that the black lines converge to the same spot for all quantities of interest. The comparison to the estimates from local data (the red lines) is not strictly necessary but can be helpful in giving insight into what additionally has been learned about the shared parameters ϕ from the new data $\bar{y}'$.

The lower-middle plot in Figure 1 shows that accuracy of the proposal distribution in importance sampling improves with more iterations and Pareto shape parameter k finally reaches values close to $\frac{1}{2}$ indicating sufficient accuracy for the approximation (see Section 1.5).

The lower-right plot in Figure 1 seems to show the efficiency of the importance ratios decreasing as the algorithm proceeds, which seems implausible. This apparent decline in efficiency is happening because (6) is only an approximate measure and tends to be overly optimistic when the number of simulation draws is small. In this simulation, we start by running Stan to produce just 100 draws of ϕ, then every update of the outer loop (that is, every 10 steps of the algorithm as shown on the plots), the number of iterations increases by a factor of $\sqrt{2}$, so that by the end we have $2^{9/2} \cdot 100 \approx 9000$ draws, which allows a better sense of the efficiency of the importance ratios. One could of course run the algorithm longer, but given the mixing of the three independent runs (as can be seen in Figure 1), this would not seem necessary.

The comparisons to the red lines (estimate from local data) and green lines (estimate from complete data) show there is information about the model parameters from $\bar{y}'$ and additionally from y' if those data were available.

Hierarchical linear model example: Posterior mean +/− sd from EP algorithm from 3 starting points
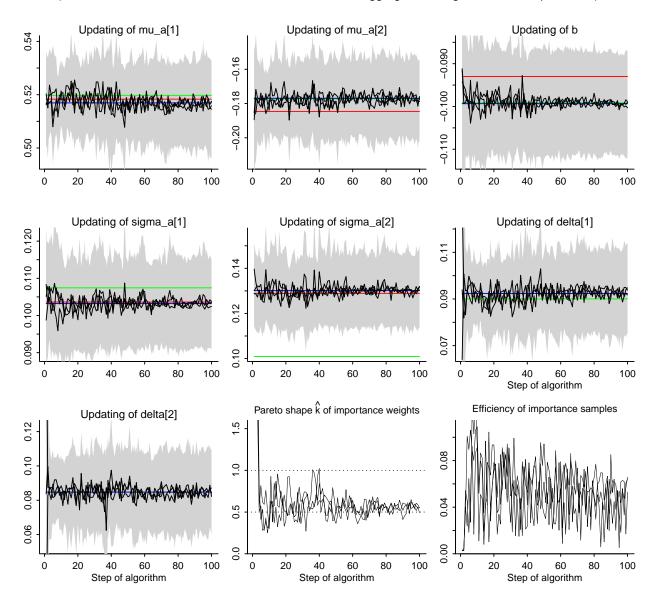(Red lines show estimate from local data, blue includes aggregate data, green uses complete data)

Figure 1: *Hierarchical linear model example: Posterior means for 3 independent runs of the algorithm (from different starting points), showing posterior mean $\pm$ standard deviation for each parameter as estimated using weighted posterior simulations, based on running the EP algorithm of section 1.5 for 100 step: 10 of the outer loop (updating $g(\phi)$) and, for each, 10 steps of the inner loop (updating $g(\delta)$). The second last plot shows the estimate the shape parameter $\hat{k}$ for the importance ratio distribution; see text for discussion of this corresponds to the accuracy of the importance sampling. The last plot shows the efficiency of the importance ratios at each step of the algorithm, as measured by the formula (6); see text for discussion of this efficiency graph.*

2.2. Hierarchical logistic regression

We repeat the above analysis, altering the example only by changing the data from continuous to binary and changing the regression model from linear to logistic, with 10 observations per cell; thus the data y_{jt} are the number of successes out of 20 tries given probability $p_{jt} = \text{logit}^{-1}(\alpha_{j1} + \alpha_{j2}x_t + \beta x_t^2)$. We keep all the parameters the same, except that the error standard deviation σ_y is no longer needed in the model.

Figure 2 shows the results. Unlike in the linear model, we cannot analytically integrate out the missing data in the external dataset, hence there is no direct way to compute the posterior distribution of the observed data $y, \bar{y}'$ and there are no corresponding blue lines on the graph showing the correct answers. We can, however, compare the three chains to each other and track convergence. We can also compare, for each parameter, to the red and green lines, which show the posterior means from the local data y and from the complete data y, y'.

The lower-middle plot in Figure 2 shows the Pareto shape parameter k decreasing during initial iterations but staying larger than 1. This indicates that the Gaussian approximations, in this logistic example, are not as good as we could wish for. The Pareto smoothing of the importance weights stabilizes the estimates and the overall algorithm converges. As the Pareto shape parameter $k > 1$, the efficiency estimate in lower-right plot is invalid and it is also likely that some of the true posterior uncertainty is underestimated.

In this particular example, the inferences for the mean parameters μ_a and the fixed coefficient b are closer to the green lines, which makes sense because both the original data y and the new data $\bar{y}'$ inform these parameters directly. In contrast, the population variance parameters σ_a are closer to the red lines, which also makes sense because our aggregate data provide almost no information about the variation among people; the green lines for σ_a represent a large amount of information that is available in the complete data y' but not in the aggregates $\bar{y}'$ and so we should not expect to attain these estimates. Finally, the parameters δ can only be estimated with the old and new data combined so there are no red lines in these graphs.

3. A differential equation model from pharmacometrics

We consider as an example a standard pharmacodynamic response model, the turn-over model (Jusko and Ko, 1994). This model is frequently used to link the drug concentration in blood plasma to some drug response, $R(t)$. The turn-over model is widely used in pharmacometrics as it is semi-mechanistic and can describe many situations while still being identifiable from sparse data. The model's physiologic interpretation is that the drug response is hypothesized to be the result of in- and out-flux from a bio-compartment which represents the drug response $R(t)$. Either the in- or the out-flux can be altered by the drug concentration which leads to a total of 4 model variants as the drug may either increase or decrease the respective flux.

By construction, the turn-over model assumes that with increasing administered drug amounts, the maximal drug effect occurs at later times. The in-flux of the bio-compartment is assumed to be of order zero, k_j^{in}, while the out-flux is assumed as first order, k_j^{out}. Here we consider the case of a stimulation of the zero order in-flux, which is defined as an ordinary differential equation (ODE),

$$\frac{dR_j(t)}{dt} = k_j^{\text{in}} \left(1 + E_{\max j} S_j(C_j(t))\right) - k_j^{\text{out}} R_j(t). \tag{7}$$

The drug effect enters this equation via the $S_j(t)$ function, which is typically chosen to be a logistic function of the log drug concentration, $C_j(t)$. At baseline $R_j(t = 0) = R_{0j}$ defines the required initial condition for the ODE. Under a time-constant stimulation, $S_j = s_j$, the ODE drives $R_j(t)$
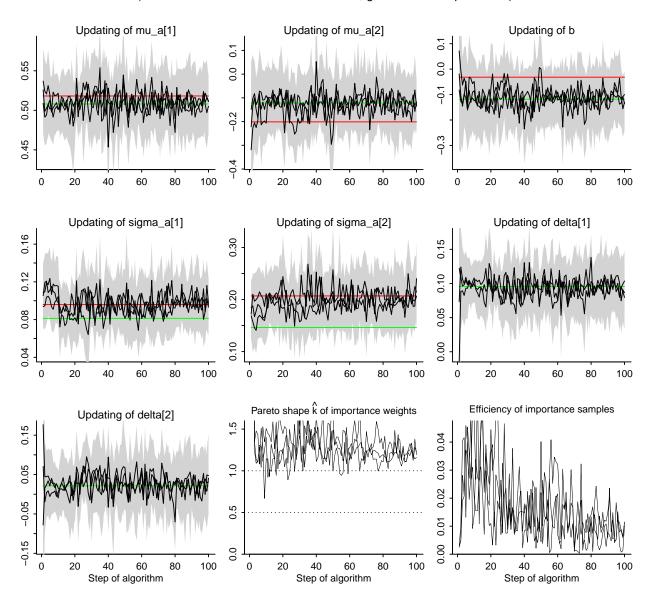
Figure 2: *Hierarchical logistic example: Posterior means for 3 independent runs of the algorithm (from different starting points), showing posterior mean $\pm$ standard deviation for each parameter as estimated using weighted posterior simulations, based on running the EP algorithm of section 1.5 for 100 step: 10 of the outer loop (updating $g(\phi)$) and, for each, 10 steps of the inner loop (updating $g(\delta)$). The second-last plot shows the estimated Pareto shape $\hat{k}$ and the last plot shows the efficiency of the importance ratios at each step of the algorithm, as measured by the formula (6); see text for discussion of this efficiency graph.*

towards it's stable steady-state which is derived form (7) by setting the left-hand side to 0,

$$R_j^{\text{ss}} = \frac{k_j^{\text{in}}}{k_j^{\text{out}}} \left(1 + E_{\max j}\, s_j\right). \tag{8}$$

Since placebo treated patients always have $s_j = 0$, the ratio $k_j^{\text{in}}/k_j^{\text{out}}$ is of particular importance in a turn-over model as it defines the response $R(t)$ in absence of drug treatment.

The function $R_j(t)$ is only implicitly defined; no closed-form solution is available for the general case of the equation (7). Without restricting the generality of our approach, we consider the case where the drug effect is for all times maximal; that is, $S_j(t) = s_j = 1$ for a patient j who receives treatment or $S_j(t) = s_j = 0$ for placebo patients otherwise. For this case, the equation can then be solved analytically to get,

$$R_j(t) = R_j^{\text{ss}} + \left(R_{0j} - R_j^{\text{ss}}\right) \exp\left(-k_j^{\text{out}} t\right), \tag{9}$$

where $s_j = 0$ for placebo patients. In the following we consider different cohorts of patients j observed at times x_t. Measurements y_{jt} of a patient j are assumed to be iid log-normal, $\log(y_{jt}) \sim N(\log(R_j(x_t)), \sigma_y^2)$. We assume that the number of patients J is large enough such that weakly-informative priors, which identify the scale of the parameters, are sufficient. The above quantities are parameterized and assigned the simulated true values and priors as:

- $J = 100$ patients in the original dataset out of which the first $j = 1, \ldots, 50$ patients are assigned a placebo treatment ($E_{\max j} = 0$) and the remaining $j = 51, \ldots, 100$ patients are assigned a treatment with nonzero drug effect ($E_{\max j} > 0$). All patients are measured at $T = 13$ time points corresponding to one measurement per month with time being measured in units of weeks, $x_t = 0, \frac{52}{12}, \ldots, 52$.

- $J' = 50$ patients in the external dataset, measured at the same 13 time points.

- $\log(R_{0j}) \sim N(l\alpha_0, \sigma_{l\alpha_0}^2)$ is the unobserved baseline value of each patient j which we set to $l\alpha_0 = \log(50)$, $\sigma_{l\alpha_0} = 0.1$. We set the weakly-informative prior to $l\alpha_0 \sim N(\log(50), 5^2)$ and $\log(\sigma_{l\alpha_0}) \sim N(\log(0.1), 5^2)$.

- $\log(k_j^{\text{in}}/k_j^{\text{out}}) \sim N(l\alpha_s, \sigma_{l\alpha_s}^2)$ is the patient-specific placebo steady state, the asymptotic value patients reach if not on treatment. In the example lower values of the response correspond to worse outcome and we set the simulated values to $l\alpha_s = \log(42)$, $\sigma_{l\alpha_s} = 0.15$ while we set the priors to $l\alpha_s \sim N(\log(50), 5^2)$ and $\log(\sigma_{l\alpha_s}) \sim N(\log(0.1), 5^2)$.

- $\log(k_j^{\text{in}}\, k_j^{\text{out}}) = l\kappa$ determines the time scale of the exponential changes. This parameterization has proven in applications to be superior in comparison to directly defining $\log(k^{\text{out}})$ as parameter. As k^{out} is a rate of change, $1/k^{\text{out}} = \tau$ is the characteristic time scale at which changes occur. Recasting $k_j^{\text{in}}\, k_j^{\text{out}}$ into $k_j^{\text{in}}/k_j^{\text{out}}\, \tau_j^{-2}$ and assuming that changes in the response happen within 10 time units led us to set $l\kappa = \log(42) - 2\log(10)$ and we defined as a prior $l\kappa \sim N(\log(50) - 2, 5^2)$.

- $\log(E_{\max j})$ is the drug effect for patient j. If patient j is in the placebo group, then $E_{\max j} = 0$. For patients receiving the assumed candidate drug, we set $\log(E_{\max j}) = lE_{\max j} = \log(0.4)$ which represents a 40% larger response score than placebo. Patients in the external data set y' are assumed to have received an alternative drug and hence are assigned a different $lE'_{\max j}$. We consider $\delta = lE'_{\max j} - lE_{\max j} = 0.2$, which corresponds to a moderate to large difference ($\exp(0.2) \approx 1.22$) in such a scenario when the two drugs are assumed to have similar effects. As priors we use $lE_{\max} \sim N(\log(0.1), 5^2)$ and $\delta \sim N(0, 5^2)$.

14

- $\sigma_y = 0.2$ is the scale of the residual measurement error on the log scale which implies that observations are recorded with a coefficient of variation equal to 20%. The prior is assumed to be $\log(\sigma_y) \sim \mathrm{N}(0, 5^2)$.

The above pharmacometric example is derived from a case presented in Weber et al. (2014). Here we now apply our procedure to simulated data with the same condition of the computations as in the previous sections. The simulated data allows for a direct comparison of results from our approximation method (y_{jt} and $\bar{y}'_t$ given) with the complete data case (y_{jt} and y'_{jt} given).

The results are shown in Figure 3. As for the proceeding logistic regression example, we cannot integrate out the missing data in the external data set such that there is again no *blue line*. We can observe that all mean estimates from the local fit, *red lines*, are altered when adding $\bar{y}'_t$. Most mean estimates ($l\alpha_0$, $l\alpha_s$ and $lE_{\max s}$) match or are very close to the estimate from the complete data situation (green lines). Only the mean estimate for $l\kappa$ is offset with respect to the green complete data line. Nonetheless, the estimate from our procedure (black lines) for $l\kappa$ is closer to the complete data case when comparing to the red line, the local only data fit. The variance component $\sigma_{l\alpha_0}$ estimate differs for all three cases (local only data, complete data, approximation method). Moreover, the variance component $\sigma_{l\alpha_0}$ is shifted from the local only data scenario case. The updating of variance components is a counter-intuitive observation since we only use aggregate mean data. This can be the consequence of correlations in the posterior which are not fully taken into account by the approximation method (δ is assumed independent of ϕ).

The lower-middle plot in Figure 3 shows that accuracy of the proposal distribution in importance sampling improves with more iterations and Pareto shape parameter k finally reaches values below $\frac{1}{2}$, indicating a good accuracy for the approximation (see Section 1.5).

The lower-right plot in Figure 3 tracks the computational efficiency of the importance sampling algorithm. We are satisfied with these efficiencies which are in the 5–20% range. Different simulation runs converge to different efficiencies, which makes sense because the approximation accuracy depends on the specific data that happen to be seen, and is not just a function of the underlying model.

4. Discussion

4.1. Simulation-based inference using an approximate likelihood

We constructed this method in response to three different issues that arose with the integration of external data into a statistical analysis:

1. Our new data were in aggregate form; the raw data y'_{jt} were not available, and we could not directly write or compute the likelihood for the observed aggregate data $\bar{y}'$.

2. The new data were conducted under different experimental conditions. This is a standard problem in statistics and can be handled using hierarchical modeling, but here the number of "groups" is only 2 (the old data and the new data), so it would not be possible to simply fit a hierarchical model, estimating group-level variation from data.

3. It was already possible to fit the model to the original data y, hence it made sense to construct a computational procedure that made use of this existing fit.

We handled the first issue using the central limit theorem, approximating the sampling distribution of the aggregate data by a multivariate normal and using simulation to compute the mean and covariance of this distribution, for any specified values of the model parameters.
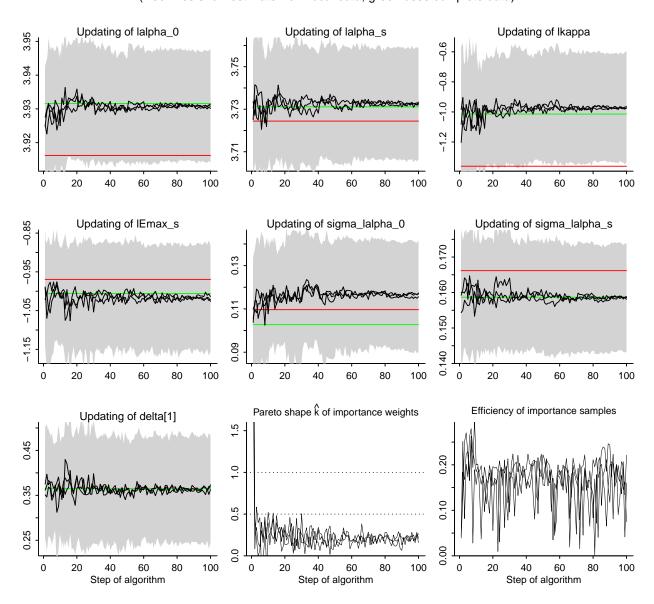
Figure 3: *Hierarchical pharmacometrics example: Posterior means for 3 independent runs of the algorithm (from different starting points), showing posterior mean ± standard deviation for each parameter as estimated using weighted posterior simulations, based on running the EP algorithm of section 1.5 for 100 steps: 10 of the outer loop (updating $g(\phi)$) and, for each, 10 steps of the inner loop (updating $g(\delta)$). The second-last plot shows the estimated Pareto shape parameter $\hat{k}$ and the last plot shows the efficiency of the importance ratios at each step of the algorithm, as measured by the formula (6); see text for discussion of this efficiency graph.*

We handled the second issue by introducing a parameter δ governing the difference between the two experimental conditions. In some settings it would make sense to assign a weak prior on δ and essentially allow the data to estimate the parameters separately for the two experiments; in other cases a strong prior on δ would express the assumption that the underlying parameters do not differ much between groups. Seen from a different perspective, the new experimental condition is considered as a biased observation of an already observed experimental condition which goes back to Pocock (1976).

Finally, we used importance sampling to make use of our initial fit, embedding the entire procedure within an expectation propagation loop so as to converge to a close approximation, thus reducing the burden on the importance-weighting correction.

4.2. Partial transmission of statistical information

We see the ideas in this paper as having broader implications in statistics, beyond the problem of including external aggregate data into a statistical analysis.

From the modeling side, the problem of imperfect aggregation or transportability arises in many application areas, not just pharmacometrics. The general setting is that a model is fit to a dataset y, giving inferences about parameter vector ϕ, and then it is desired to use ϕ to make inferences in a new situation with new data y'. The often-voiced concern is that the researcher does not want the model for y' to "contaminate" inference for ϕ. There is a desire for the information to flow in one direction, from y to ϕ to predictions involving the new data y', but not backward from y' to ϕ. Such a restriction is encoded in the `cut` operator in the Bayesian software Bugs. We do not think this sort of "cutting" makes sense, but it arises from a genuine concern, which we prefer to express by modeling the parameter as varying by group. Hence we believe that the introduction of a shift parameter δ, with an informative prior, should be able to do all that is desired by the `cut` operator. Rather than a one-way flow of information, there is a two-way flow, with δ available to capture the differences between the two groups, so there is no longer a requirement that a single model fit both datasets.

4.3. Expectation propagation as a general adjunct to importance sampling

From the computational perspective, importance sampling is used in a wide variety of settings. Our iterative EP algorithm has the potential to make importance sampling practical in a much broader range of scenarios: the EP step allows much of the information in the importance weights to be entered into the model as a pseudo-prior, thus reducing the variance of the importance weighting that remains. This is a quite general idea, not restricted to hierarchical models or aggregate data, and should be implementable just about anywhere that importance sampling is done.

References

Dominici, F., G. Parmigiani, R. L. Wolpert, and V. Hasselblad (1999). Meta-analysis of migraine headache teatments: Combining information from heterogeneous designs. *Journal of the American Statistical Association 94* (445), 16–28.

Gelman, A. (2004). Parameterization and Bayesian modeling. *Journal of the American Statistical Association 99* (466), 537–545.

Gelman, A., J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin (2013). *Bayesian Data Analysis* (Third ed.). Chapman & Hall/CRC.

Gelman, A. and D. B. Rubin (1992). Inference from iterative simulation using multiple sequences. *Statistical Science 7*(4), 457–472.

Gelman, A., A. Vehtari, P. Jylänki, C. Robert, N. Chopin, and J. P. Cunningham (2014). Expectation propagation as a way of life. *arXiv:1412.4869 [stat]*. arXiv: 1412.4869.

Geweke, J. (1989). Bayesian inference in econometric models using Monte Carlo integration. *Econometrica 57*(6), 1317–1339.

Higgins, J. P. T. and A. Whitehead (1996). Borrowing strength from external trials in a meta-analysis. *Statistics in Medicine 15*(24), 2733–2749.

Ionides, E. L. (2008). Truncated importance sampling. *Journal of Computational and Graphical Statistics 17*(2), 295–311.

Jusko, W. J. and H. C. Ko (1994). Physiologic indirect response models characterize diverse types of pharmacodynamic effects. *Clinical Pharmacology and Therapeutics 56*(4), 406–419.

Marin, J.-M., P. Pudlo, C. P. Robert, and R. J. Ryder (2012). Approximate Bayesian computational methods. *Statistics and Computing 22*(6), 1167–1180.

Minka, T. P. (2001). Expectation propagation for approximate Bayesian inference. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, UAI'01, San Francisco, CA, USA, pp. 362–369. Morgan Kaufmann Publishers Inc.

Opper, M. and O. Winther (2000). Gaussian processes for classification: Mean-field algorithms. *Neural Computation 12*(11), 2655–2684.

Opper, M. and O. Winther (2005). Expectation consistent approximate inference. *Journal of Machine Learning Research 6*, 2177–2204.

Pitt, M. K., M.-N. Tran, M. Scharth, and R. Kohn (2013). On the existence of moments for high dimensional importance sampling. *arXiv:1307.7975 [stat]*. arXiv: 1307.7975.

Pocock, S. J. (1976). The combination of randomized and historical controls in clinical trials. *Journal of Chronic Diseases 29*(3), 175–188.

Stan Development Team (2015). Stan: A C++ library for probability and sampling.

Vehtari, A., A. Gelman, and J. Gabry (2015). Pareto smoothed importance sampling. *arXiv:1507.02646 [stat]*. arXiv: 1507.02646.

Weber, S., B. Carpenter, D. Lee, F. Y. Bois, A. Gelman, and A. Racine (2014). Bayesian drug disease model with stan: Using published longitudinal data summaries in population models. Alicante, Spain.

Zhang, J. and M. A. Stephens (2009). A new and efficient estimation method for the generalized Pareto distribution. *Technometrics 51*(3), 316–325.